Optimized SPARQL performance management via native API

Date: February 27, 2014.
Team lead: Victor Chernov
E-Mail: vchernov@nitrosobase.com
Time Zone: MSK, UTC+4

Hackathon
Ontology Summit 2014
The Goals

• Studying the kinds of queries revealing the advantages of one or another database
  – Selection of a SPARQL subset from SP2Bench test, dataset and loading it to all triple-stores.
  – Implementing measurement aids, testing
  – Accurate time measurement, getting min, max, average and median times.
  – Reflection on the results, advantages and disadvantages of the triplestores on each selected query.
Triplestore selection

• We are going to compare the following triplestores:
  – Virtuoso
  – Stardog
  – NitrosBase

• The triplestores have the following important advantages:
  – Very high performance on sp2bench benchmark
  – Linux and Windows versions
  – Native API for fast query processing
Writing codes for fast query execution

• We are going to use native API for fast query execution
  – **Virtuoso** provides Jena, Sesame and Virtuoso ODBC RDF Extensions for SPARQL
  – **Stardog** provides the core SNARL (Stardog Native API for the RDF Language) classes and interfaces
  – **NitrosBase** provides C++ and .NET native API
• Writing additional codes needed for testing:
  - Accurate time measurement;
  - Functions for getting min, max, average and median times;
  - Functions for getting time of scanning through the whole query result;
  - Functions for getting time of retrieving first several records (for example, the first page of web grid);
  - Etc.
Loading test dataset

- Selecting a data subset from sp2bench benchmark;
- Measuring data loading time;
- Data are considered as loaded as soon as the system is ready to perform a simplest search query. This is done to eliminate background processes (eg. indexing).
Preparing SPARQL queries

• We are going to explore the query execution performance by the databases under consideration.

• The queries should be fairly simple and cover the different techniques, for example:
  - search the small range of values
  - search the big range of values
  - Sorting
  - Aggregation
  - Several different join queries
  - Retrieving part of result
  - Retrieving whole result
  - etc.
Testing, reflection, reporting

• **Testing by participants**

**Note:** During testing each database may allocate a lot of resources, that can affect the performance of other databases. That’s why each test should be started from system reboot.

• **Reflection on results**

• **Report writing**
9. Contacts

Victor Chernov
vchernov@nitrobase.com
+7(985)999-22-43